



Yan, Y., Farhadi Beldachi, A., Nejabati, R., & Simeonidou, D. (2020). P4-enabled Smart NIC: Enabling Sliceable and Service-Driven Optical Data Centres. *Journal of Lightwave Technology*, 38(9), 2688 - 2694. <https://doi.org/10.1109/JLT.2020.2966517>

Peer reviewed version

Link to published version (if available):
[10.1109/JLT.2020.2966517](https://doi.org/10.1109/JLT.2020.2966517)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IEEE at <https://ieeexplore.ieee.org/document/8959326>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: <http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

P4-enabled Smart NIC: Enabling Sliceable and Service-Driven Optical Data Centres

Yan Yan, Arash Farhadi Beldachi, Reza Nejabati, Dimitra Simeonidou

Abstract—This paper reports an FPGA-based P4-enabled Smart NIC solution which is designed and implemented for web-scale cloud and to meet 5G/Beyond 5G networking requirements. The P4-enabled Smart NIC solution leverages the open standards, platforms and software-defined approaches, responds to the real time Data Centre Networking service requests, in particular, enables the end-to-end network slicing, which is one of the critical requirements of multi-tenancy 5G network. We discussed the possibilities and challenges of P4 specification implementation in the FPGA to realise the Smart NIC functionalities. And after that, we showed its data plane programmability and flexibility with P4 features. Furthermore, we demonstrated its application scenario in an 5G environment mainly focusing on edge Data Centre to core Data Centre network slicing. The setup interconnects the P4-enabled Smart NIC with optical Bandwidth Variable Transponders, and the system offers agile 100Gbps interface to transport the packets through P4-defined data plane for L2/L3/L4 parsing and action. The P4-enabled Smart NIC can change the data plane pipelines in seconds, and it can achieve maximum 84.8Gbps throughput. With P4 programmed hardware offloaded Segment Routing can produce 30% more bandwidth than without.

Index Terms—P4, Smart NIC, Network Slicing, 5G

I. INTRODUCTION

LARGE scale 5G deployments by major operators globally, along with advances in Augmented Reality (AR), Virtual Reality (VR), Internet of Things (IoT), self-driving vehicles[1], have been promising enough to make network technologists and innovators to start the discussions what will be the next generation (i.e. Beyond 5G) of the network. After decades of stifling innovation in the telecoms industry, due to monopolisation, technological sophistication, and lack of competitiveness in the market, things changed when web scale companies demanded cheaper and more flexible technologies. Its needed to democratize networking technologies in order to accelerate the innovation and creativity. Therefore, a number of open standards and protocols, along with reference implementations were introduced. Software Defined Networking (SDN) controllers with the flagship Openflow [2] has been one of the major attempts in generalising and

standardising network programming. And actually, the SDN or centralized controller is being rolled out in the network or under the hood to enable orchestrated systems. Network Function Virtualization (NFV) with virtualised network functions, nowadays implemented in the communication service providers, are mainly legacy software loaded onto VMs. However, virtualized functions and elements, which on-demand adapt the network resources to the required quality of service and performance level [29] are going to be one of the main enablers for the next generation of the networking technologies.

In view of 5G network, the diversity of applications and its Key Performance Indicators (latency, bandwidth, synchronization, and etc.) requires the multiplexing capability of virtualised and independent logical network functions on the same physical network infrastructure. To support varieties of functional split, which in some cases, can be dynamically and flexible allocate the functions and resources, approaches such as transport network slicing play a key role on the 5G multi-tenancy model [3]. Network slicing across multiple layers, i.e optical, IP, and applications is a critical feature in 5G networks and beyond. Its virtualised technology framework allows tailoring the network performance (latency and throughput) and functionality to the tenants' (mobile operator, DC applications, fintech, and so on) requirements. It enables forwarding the packets through an ordered list of instructions, and dynamically demanding the bandwidth and latency.

In the IP domain, Segment Routing (SR) is one of the main candidates in providing virtualised layer 2 and Layer 3 Virtual Private Networks (VPNs). It utilises source routing and as an enhancement to Multi-Protocol Label Switching (MPLS), has been designed to with central controller in mind for label assignment and distribution [4]. Various efforts have demonstrated source routing from servers to send traffic all the way through the core and finally to the destination [5].

For optical networking, various vendors including the web scale internet companies, such as Facebook, have been developing compact optical transport systems [6] known as Data Center Interconnect (DCI), which are available in pizza box sizes such as IP/Eth solutions. They offer multi-rate multi-protocol client ports 10Gbps to 100Gbps, and variable

This paper was submitted on 15th November for review.

Yan Yan is with the Raymax Technology Ltd., Hangzhou, China. She is also a PHD candidate with the High Performance Networks Group, the Electrical Engineering Department, University of Bristol, Bristol, UK. (e-mail: yan.yan@raymax.net).

Arash Farhadi Beldachi is with the High Performance Networks Group, the Electrical Engineering Department, University of Bristol, Bristol, UK. (e-mail: Arash.Beldachi@bristol.ac.uk).

Reza Nejabati is with the High Performance Networks Group, the Electrical Engineering Department, University of Bristol, Bristol, UK. (e-mail: Reza.Nejabati@bristol.ac.uk).

Dimitra Simeonidou is with the High Performance Networks Group, the Electrical Engineering Department, University of Bristol, Bristol, UK. (e-mail: Dimitra.Simeonidou@bristol.ac.uk).

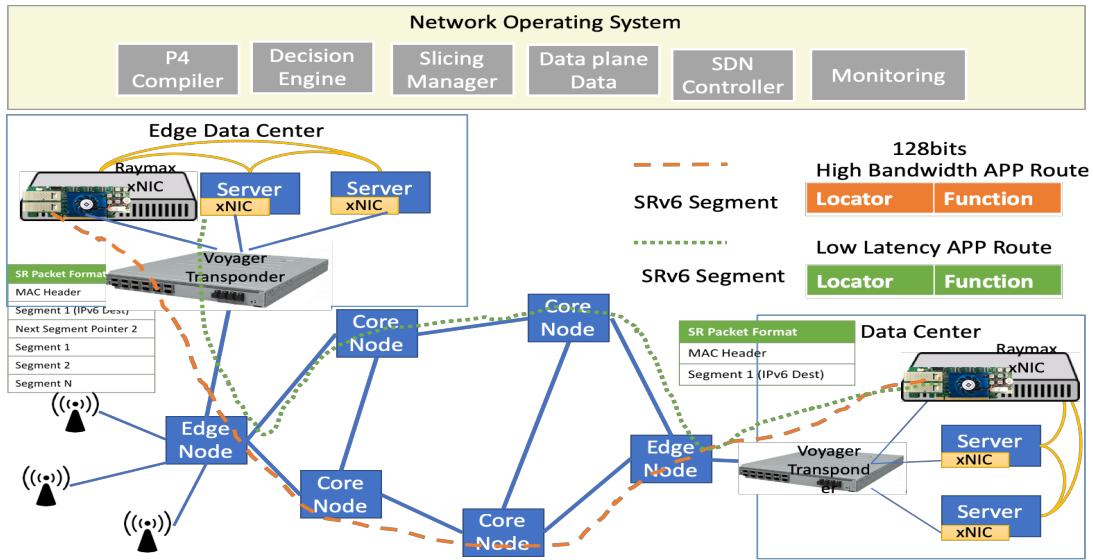


Fig. 1. Segment Routing Transportation in 5G network Edge Data Centre to Core Data Centre Architecture

bandwidth allocation Bandwidth Variable Transponder (BVT) towards the core, where coherent technologies can take the signal for much longer distances.

A. DC architecture evolution and new open initiatives

With DCs become flatter in architecture to accommodate east west traffic patterns, optical interconnects are getting closer to the network edge to offer higher bandwidth/cost efficiency, making servers with NICs becoming the network edge where policies, Quality of Service (QoS) will happen before traffic exists the server.

When communicating and controlling network equipment, the boxes which actually forward the packets and bytes in the network, essentially limit how the software can define them. There has been a number of attempts to introduce open hardware platforms [7]. Recently P4 [8] consortium has emerged as a result of research community efforts in providing a means to define packet processing pipelines on the fly. It has to be mentioned this consortium and the available standards and protocol are still in its infancy, and that makes efforts in utilising this technology even more worthwhile so help shaping up its progress. Companies such as Barefoot networks [9] are a proponent of bringing open standards to data plane. In addition, more recently major networking chip providers have started to introduce P4 capabilities as the approach is gaining more tractions [10] [11].

Additionally, a number of open design and standard forums initiated by Linux Foundation [12], and some others such as Open Compute, FD.io [13], have been formed to provide free design blueprints for various segments of IT and Networking industries, to boost the community contribution as well as facilitating the integration of various technologies in IT with networking to achieve higher efficiency and performance.

B. NIC Technology Background

The Network Interface Card (NIC) plays a key role on the server edge to communicate with the outside. Network processing in low to medium speeds (<10G) is still well possible exploiting native operating system (OS) drivers and powerful multicore processors available in the market.

However, to achieve very high bit rate (>100G), servers need to use technologies such as kernel bypass and offloading to be able to reach the high throughputs without using the CPU cycles out of proportion. Offloading various processing and network functions to the NICs, such as SSL and TCP/IP offload [14], can free up the CPU from some of the network stack and therefore improve the overall performance on the server. On system level, introducing Single-root input/output virtualization (SR-IoV) [15] allows the NIC accessing the Virtual Machines (VMs) directly with no interrupt of the CPU, which decreases the latency among VMs and increases the bandwidth. Introducing SDN applications and infrastructure (such as vSwitches) to the server, demands extra load on the hardware to provide a very dynamic environment for their operations. Programmable data planes, both virtual and physical, can optimise the network and processing utilisation, which is one of the main goals of introducing P4 enabled Smart NICs. Additionally, Smart NICs can help this new requirement to provide a hosting offload environment so to help with compute as well as communications. The P4 enabled Smart NIC solutions, offering programmable networking and compute, needs to be integrated with the available acceleration technologies such as DPDK [16] and NetMAP [17] to achieve end to end optimum results.

In today's market, there are basically three types of Smart NICs, the ASIC-based, the Multicore NP/SoC-based, and the FPGA-based. Comparing these three types of Smart NICs [18], considering the dynamic applications in the network, the requirement on P4-enabled, and the time-to-market, FPGA is a promising solution by providing the programmability, nano second processing time, and performance and efficiency of customised implementation. The most recent FPGA chipsets are equipped with high capacity network and storage IO and resources which makes them extremely adaptive and responsive to quickly respond to the network requirement and off-load the CPU. Azure Accelerated Networking with FPGA-based Smart NIC [19] have already showed its successful use case at hyperscale cloud.

In the paper, we focus on the design, implementation, and validation of P4-enabled Smart NIC. As the nature of the NIC

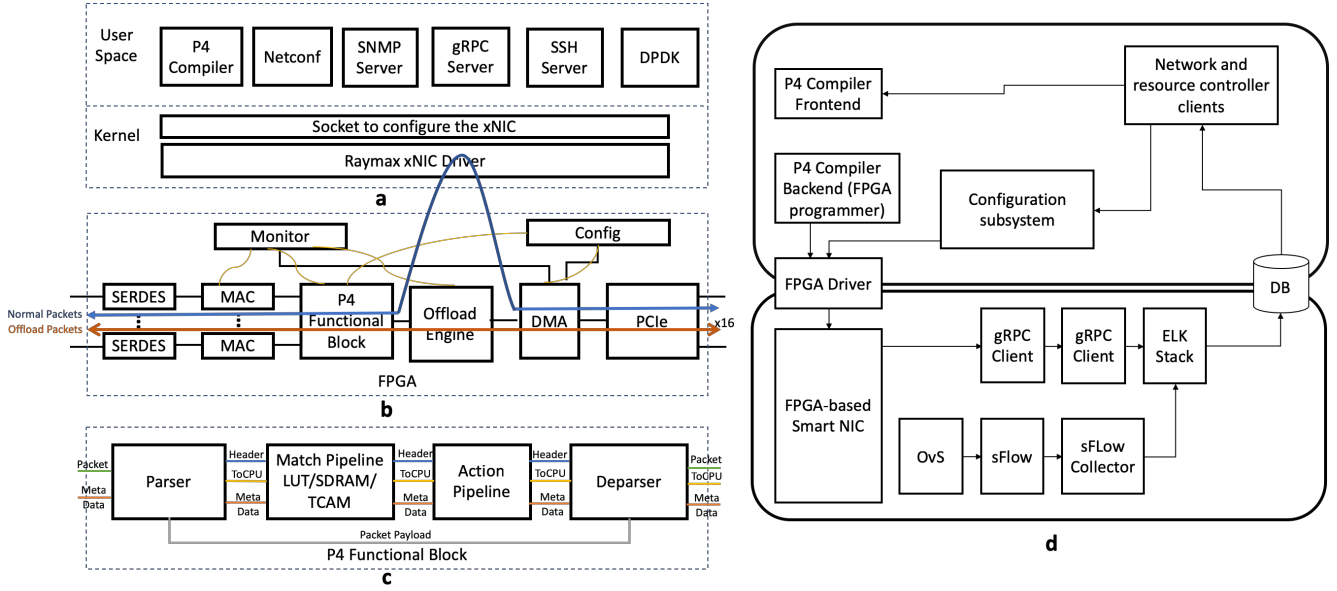


Fig. 2. Smart NIC Architecture: (a) Software Architecture, (b) FPGA-based High Level Architecture, (c) FPGA-based P4 Functional Block Architecture, (d) Hardware/Software control flow

is plugged into the server to adapt to both hardware and OS, we demonstrate the software side and FPGA-based hardware side of the design. P4 supplies the programmability and flexibility to the Smart NIC, which allows tackling network operations without interpreting CPUs. In the scenario of 5G network, we showed a use case of inter-DC network slicing and the experimental results.

II. RAYMAX SMART NIC ENABLED OPTICAL 5G INTRA-DC AND INTER-DC ARCHITECTURE

The proposed optical 5G inter-DC architecture in Fig.1, shows a converged fronthaul and backhaul network architecture, where virtualization and data plane programmability are key enablers. The Raymax FPGA-based Smart NIC with P4 features is the main enabler for the proposed data plane programmability. The Voyager BVT Transponder enables the programmability in the optical domain.

A. Raymax SmartNIC enabled intra-DC architecture

The P4-enabled Smart NICs, plugged into the servers, enable intra-rack server-to-server full mesh direct connection, which eliminate the electronics in the Top of Rack (ToR), allowing a pure optical ToR (i.e spectrum selective switch, wavelength selective switch) [20] or a DCI to directly transmit over long distance. The full-mesh interconnect with other intra-rack servers allows the servers in the same rack communicate with each other without going through the ToR switch, which saves more than 80% of the intra-rack link latency (considering the ToR switch latency).

B. Raymax SmartNIC enabled inter-DC architecture

We were focusing on the edge DC to core DC end-to-end network slicing via segment routing. The Segment Routing (SR) IPv6 headers and MPLS labels can be inserted directly in the smart NIC by compiling the P4 files and downloaded to the Smart NIC through the socket. Compared to inserting the SR headers by the servers or the software, the P4 supplies better programmability and its enabled NIC offers better network

performance and less CPU utilisation. Fig.1 as an SRv6 example, displayed the way of by inserting diverse segment identifications, the packets can go through the routes by its segments only to a high bandwidth route (slashed) or a low latency route (dotted). When reaching to the end, the segments will be deleted by the end point server Smart NIC.

III. FPGA FOR P4-ENABLED SMART NIC ARCHITECTURE AND DESIGN

The system design includes software part in Fig.2(a) and FPGA-based data plane part in Fig.2(b) and Fig.2(c). The hardware and software control flow blocks are in Fig. 2(d).

A. The Goal of the P4-enabled Smart NIC Solution

Our solution stretches in the open networking domain, leveraging open source platforms and software-defined standards, such as Software Defined Network (SDN), FD.io, P4 and so on, we built our innovated white box solution from the software to the hardware.

With 100Gbps line cards becoming a new default server networking requirement, we have prototyped the Smart NIC to fit the multiple networking and computational requirements. Our goals of the system are as following:

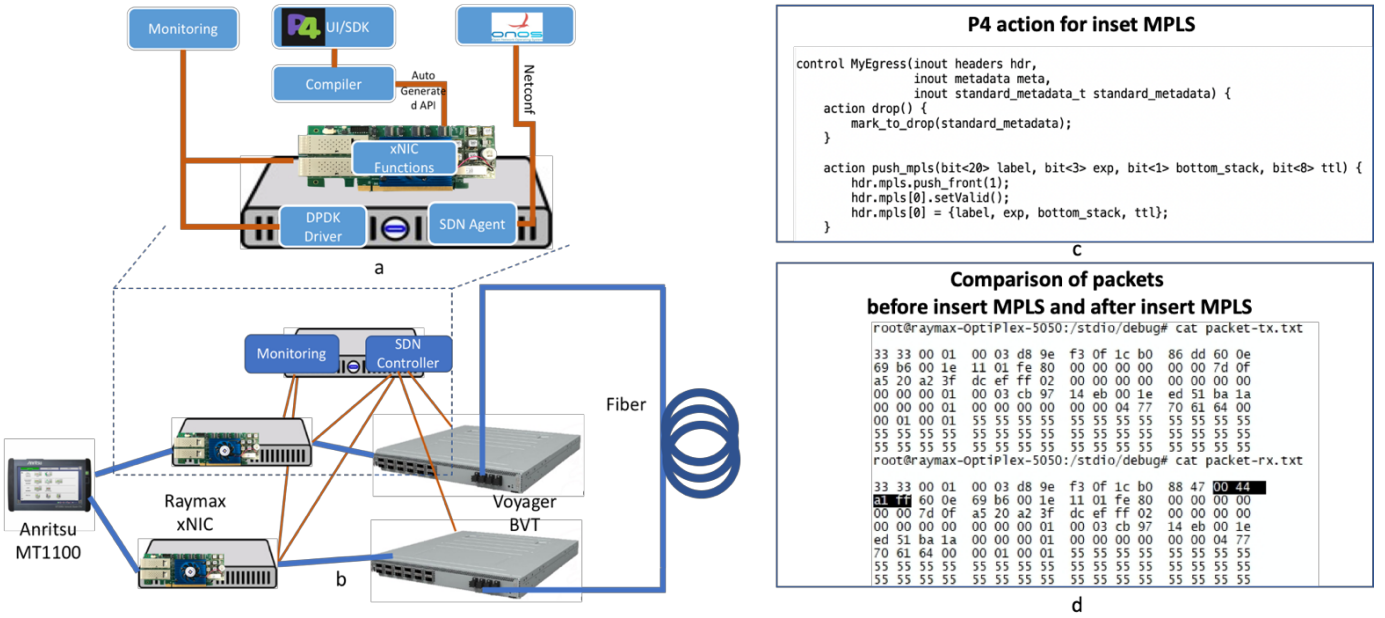


Fig. 3. (a) Smart NIC setup, (b) Testbed setup, (c) P4 action for insert MPLS, (d) Comparison of packets before insert MPLS and after insert MPLS

1) P4-enabled flexible dataplane

To realise the flexible and programmable data plane with pipelined data path.

2) Offload CPU

Allowing the Smart NIC process the network services and stacks, save the CPU for more application-based processing.

3) Intent driven networking

To seamlessly deploy and accelerate new applications by their intended networking and computational requirements using

4) Inband network telemetry [21]

To collect and report the network state, such as the throughput, latency, congestion status, resource utilization and etc., by the data plane itself.

B. The Software Architecture and Design

We developed the kernel driver and DPDK driver to accommodate our FPGA-based Smart NIC. In addition to the kernel layer, we tried to use the open networking projects and existing protocols, i.e. P4C, ONOS and gRPC, shown in Fig.2(a), to provide data and control access to Smart NIC environment. We used our own configuration graphical user interface (GUI) for updating flows and rules. However, we implemented the northbound API for the future SDN controller integration, and the OvS (Open virtual Switch) plays the role as an SDN agent for flow translation to the Smart NIC. In addition, we built our own P4 agent after the P4C to fit to our backend device. While the majority of the offload and low latency work were implemented in the FPGA, the software is mainly responsible for supporting the communication between FPGA and server.

C. The FPGA-based Architecture and Design

On the programmable data plane, to achieve the 100Gbps bandwidth, our prototype is targeting on the Xilinx Ultrascale+ series FPGA with 16*Gen3 PCIe interface, 100Gbps QSFP28 transceivers, Samtec Firefly 25G*8 optical transceivers.

The FPGA-based data plane part (Fig.2(b)) is implemented

mainly for enabling the traffic flow between the SERDES and PCIe. The main functional block is P4 functional block and offload engine. With the normal, no offloading NIC, the traffic needs to be iterated processed by CPU, and then sent to the NIC; in contrast, with the Smart NIC and P4-enabled offload engine, the traffic matches and implements the actions in the FPGA, which is paralleled processed.

The implementation of the P4 data plane block (Fig 2(c)) in the FPGA was to follow the P4_16 language specification [22], and processed the packets as required. Although the standard still is young, however we are leveraging on this to offer enhanced interoperability. The hardware capabilities nonetheless can be used regardless of the standard. The P4 data plane functional block can be separated to parser, match, action and deparser. The packets processing was pipelined, while the packets were parsed, the metadata were extracted and matched with the mask table and matching table. Afterwards, the filtered packets followed the action rules, and then got repacked to a new packet for output. The parser was designed with two modes: one is the full parser mode that can parse the header of Ethernet, IPv4/IPv6, UDP, TCP, VxLAN, VLAN (3 nested), and MPLS (3 nested); the other one is the byte-based mode that can get the information from the P4 agent byte by byte on what to parse and how to parse.

The other major function blocks include the matching block and action block. For the matching block, considering the nature of P4 language and its converted binary mask file, to achieve the fast searching and matching, we employed TCAM [23] to match the mask key and search key to search for the data in the specific address with hit/miss indication. The implemented match block supported 10 to 660 mask keys and 32 matched output. Considering the match state '1', '0', 'X', it can support up to 3^{660} matching cases. The matching block can be cascaded for the use case with priority requirement of more than 32 output requirements.

For the action blocks, to realise the P4 standard set of primitive actions, we combined some similar actions, like a

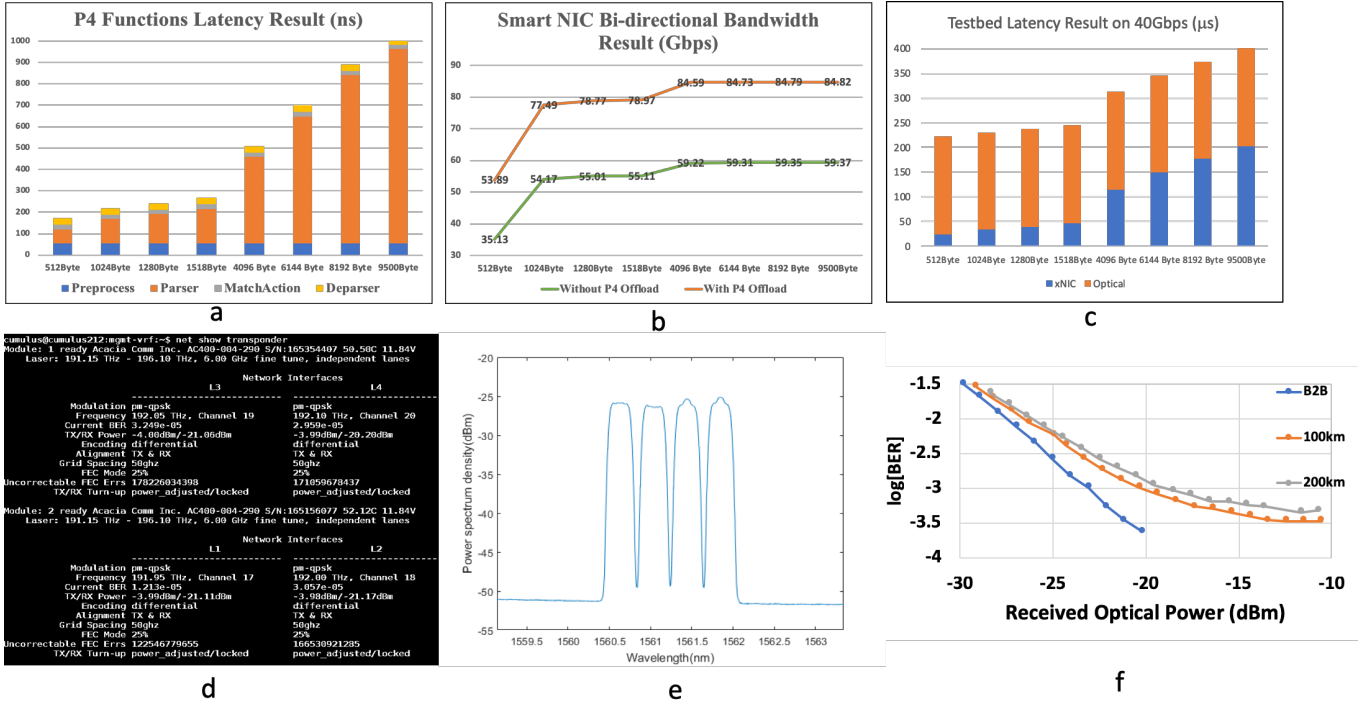


Fig. 4. (a) P4 functions Latency Result, (b) Smart NIC bandwidth result, (c) Testbed latency result, (d) Voyager BVT setup, (e) Spectrum result, (f) Long haul BER.

group of bit operation, grouped modify and set, and etc. to save the logic utilisation. When the header was modified, the checksum was pipelined and calculated afterwards.

D. The FPGA-based Implementation Beyond P4

Considering the power of the FPGA is far beyond just for network processing, current applications, such as machine learning, edge computing, and cryptos all have successful solutions using FPGA [24] [27]. We considered using FPGA-based Smart NIC as an application-driven server edge processor.

It is well known that the FPGA allows fully reconfiguration by downloading the bit files to realise completely different functions. Besides, Partial Reconfiguration technology, by using the partial reconfiguration controller IP, allows the designer to download partial bit files while the remaining logic continues to operate without interruptions. This method allows the application to change the hardware functionality on the fly, which enhancing the flexibility that FPGA-based implementation offer.

By utilizing partial reconfiguration technology, we designed and implemented P4 ACTION functional block as a partial reconfigurable block that can meet the service requirement on the fly. When the use cases need additional sets of actions or processing or algorithms, the actions can be hitless partial reconfigured without stopping the service. The same as supporting other protocols. Thus, our Smart NIC P4 data plane could act as an application driven enabler in the Data Center Networking (DCN).

E. Hardware and software control flow

The hardware and software control flow blocks are demonstrated in Fig.2(d). The clients got the new network requirements and send them to the Smart NIC with the network controller. To program the network data plane, the clients need

to write their own P4 files, get it compiled and send it to the FPGA-based Smart NIC through the FPGA driver; in the meantime, the controller extracts the configuration data, got it filled by the clients and send it to the Smart NIC through the FPGA driver as well. The FPGA-based Smart NIC generates the performance result (latency, bandwidth) per flow and transmits the result through gRPC client and server tunnel to the ELK Stack, a collection of opensource products for logging and searching. It is also employed here for flow information collection from OvS.

IV. TESTBED SETUP AND RESULTS

A. Testbed setup

We set up a Smart NIC testbed with control plane and data Plane in Fig.3 (a). The Smart NIC was plugged into the server, and the server was installed with DPDK driver, SDN agent and ONOS controller. The P4 files could be pasted or uploaded to the SDK/UI, which would compile and translate the P4 files to the Smart NIC. There was a monitoring module for collecting and displaying the statics from Smart NICs and servers.

Furthermore, to emulate the edge-DC to DC environment, and displayed the IP domain segment routing combined with optical domain network slicing. We set up the optical testbed as displayed in Fig.3 (b) with Voyager BVT Transponders and back-to-back fibres. To measure the latency and bandwidth results, we employed Anritsu MT1100 traffic analyser to generate and analyse the result. The testbed was set up to assign different wavelengths to SR-MPLS labels, which enables server-to-server, end-to-end network slicing in both IP and optical domain. We wrote a .p4 file, with a functionality that insert MPLS-SR header in ingress and delete in egress. Fig.3 (c) shows the P4 action for insert SR-MPLS, and Fig.4 (d) gives the captured packet after inserting the SR-MPLS.

B. FPGA-based Resource Utilization

The challenge of the parse-match-action was the balance of the timing closure and the combinational logics. We managed to achieve 350MHz clock frequency, and the minimum utilisation of the FPGA resources were shown in Table I. A few tools had already been existed for converting P4 files to FPGA bit files, such as P4-SDNet [25] and P4FPGA [26]. However, since these tools performed the kind of high level language to register transfer level (RTL) translation, which were not resource and performance optimised as RTL level input directly, the resource utilisation results were not comparable.

As shown in Table I, the resource block RAMs(BRAMs) are mainly used for ternary content-addressable memory (TCAM) match and configure block, the flip flops (FFs) are mainly for parser, and look-up-tables (LUTs) are mainly used on the action block for the combinational logic. Based on the application and the requirement, the design can be extended to integrate more match-and-action stages and matching more rules.

TABLE I. P4 BLOCK RESOURCE UTILIZATION

resource	utilization	available	utilization %
LUT	38855	394080	9.85
LUTRAM	4292	197280	2.17
FF	9560	788160	1.21
BRAM	120.5	720	16.7

C. P4 agility result

Our P4 compiler translates the .json file (compiled from .p4 file) to the P4 binary file that FPGA can understand. After the compiler, the .p4 file was translated to a mask table, a match table and an action table, which were sent to FPGA through PCIe. We measured the latency from click the P4 compile button till the smart NIC driver gets the translated binary file and is about to send to FPGA through PCIe. To avoid the system time inaccuracy, we wrote a script to run the tests 100 times and measured the whole latency. The latency result is 3.07s, which proves the network manager can change the data plane behaviour in seconds. It is much lower compared to current Xilinx's P4-SDNet solution, which takes hours to complete the same functions.

D. Testbed Experimental Result

The measurement results were shown in Fig.4. We measured the P4 block latency in the FPGA (as in Fig.4(a)). The latency result revealed the detail latency of each functional block, as demonstrated in Fig.4 (a), the parser's latency was determined mainly by the Ethernet frame length, while other functional blocks' latency is comparable fixed.

The bandwidth results in Fig.4 (b) was measured with 1 CPU core in the PC of Intel Core i7-7700K CPU @ 4.20GHz x8, 62.8GiB Memory hardware setup. We inserted one SR-MPLS header to the packet and measured the maximum bandwidth of inserting by software (Without P4 SR-MPLS Offload) and inserting by FPGA (With P4 SR-MPLS Offload). The result demonstrated, with offload, the Smart NIC was able to achieve

maximum 78.97Gbps throughput with 1518 Ethernet frame size and could go up to 84.82Gbps with jumbo frame size(9500 Bytes). Without offload, the bandwidth went down maximum 30%.

The whole testbed latency was planned to be measured by Anritsu MT1100, however, the MT1100 we used, only has one 100Gbps Ethernet port, but Voyager BVT needs 2*100Gbps ports to setup. Therefore, we measured the latency separately with MT1100 to NICs, then NICs to optical devices and fibers. The whole testbed latency in Fig.4(c) demonstrates in segments of the latency of smart NIC and optical devices.

Fig.4 (d) and (e) displayed optical results. We setup the Voyager BVT with Quadrature Phase Shift Keying (QPSK) modulation format. The frequencies are set as 191.95THz, 192.00THz, 192.05THz, 192.10THz, each of them matched to a MPLS segment enabling the optical domain slicing, and the spectrum result is in Fig.4 (e). Regarding to the spec of Voyager BVT, with longer distance or higher bandwidth requirement, the modulation format can be tuned to 8QAM or 16QAM. Fig.4 (f) displays the BER for the long-haul network scenario, where the worst-case scenario corresponds to the voyager modulation of 16QAM. The increased BER corresponding to the 100km and 200km cases are due to the amplifier noise and the dispersion imposed by the optical fibre transmission [28].

V. CONCLUSIONS

In this paper, we propose an optical 5G inter-DC architecture powered by P4-enabled Smart NIC, which enables sliceable and service-driven inter-DC communication. We demonstrated the design, implementation and experimental results of P4-enabled smart NIC and its enabled inter-DC network slicing. In the experiment, with Voyager BVT, we were able to show the smart NIC's capabilities on enabling inter-DC end-to-end network slicing in both IP domain and optical domain. The measured results showed the Smart NIC can achieve maximum 84.8Gbps utilising only one CPU core. With P4 SR-MPLS Smart NIC header insertion, the bandwidth performance can be up to 30% higher than without.

REFERENCES

- [1] Y. Yan, T. Shen, A. Beldachi, K. Rajkumar, R. Wang, R. Nejabati, D. Simeonidou, "P4-enabled Smart NIC: Architecture and Technology Enabling Sliceable Optical DCs", in Proc. ECOC, Dublin, Ireland, 2019.
- [2] "OpenFlow Switch Specification," Open Networking Foundation, 2014.
- [3] K. Sparks, M. Sirbu, J. Nasielski, L. Merrill, K. Leddy, and etc. "5G Network Slicing Whitepaper".
- [4] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018
- [5] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2429-2453, third quarter 2018
- [6] G. Robers, "TIP Open Optical Packet Transport-A game-changer for R&E networking," in 9th CEF Networks Workshop, 2017
- [7] 'Open Compute Project. OCP is reimagining hardware.' <https://www.opencompute.org/>, accessed 02 April 2019
- [8] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford et al., "P4: Programming Protocol-independent Packet Processors," SIGCOMM Comput. Commun. Rev. 44, 3, July 2014
- [9] Barefoot Tofino. P4-programmable Ethernet switch ASICs. <https://barefootnetworks.com/products/brief-tofino/>

- [10] "P4-SDNET user guide", Xilinx, 2018.
- [11] Broadcom Trident 3. High-Capacity StrataXGS Trident 3 Ethernet Switch Series. <https://www.broadcom.com/products/ethernet-connectivity/switching/strataxgs/bcm56870-series/>
- [12] Linux Foundation. <https://www.linuxfoundation.org/>
- [13] FD.io. The fast data project. <https://fd.io>
- [14] Microsoft. TCP/IP offload. <https://docs.microsoft.com/en-us/windows-hardware/drivers/network/tcp-ip-offload>
- [15] Microsoft. Overview of Single Root I/O Virtualization. <https://docs.microsoft.com/en-us/windows-hardware/drivers/network/overview-of-single-root-i-o-virtualization--sr-iov->
- [16] DPDK. Data Plane Development Kit. <http://dpdk.org/>
- [17] Netmap. The Netmap project, the fast packet I/O framework. <http://info.iet.unipi.it/~luigi/netmap/>
- [18] A. Caulfield, P. Costa, M. Ghobadi, "Beyond SmartNICs: Towards a Fully Programmable Cloud," In Proc. of IEEE HPSR, 2018.
- [19] Daniel Firestone, Andrew Putnam Sambhrama Mundkur Derek Chiou Alireza Dabagh, Mike Andrewartha et al., "Azure Accelerated Networking: SmartNICs in the Public Cloud," in NSDI'18, 2018.
- [20] Y. Yan, G. M. Saridis, Y. Shu, B. R. Rofoee, S. Yan, M. Arslan et al., "All-Optical Programmable Disaggregated Data Centre Network Realized by FPGA-Based Switch and Interface Card," in Journal of Lightwave Technology, vol. 34, no. 8, pp. 1925-1932, 15 April 15, 2016.
- [21] Changhoon Kim, Parag Bhide, Ed Doe, Hugh Holbrook, Anoop Ghanwani et al., "In-band Network Telemetry," June 2016
- [22] P4. P4 Specification. <http://p4.org/spec/>
- [23] "Ternary Content Addressable Memory (TCAM) Search IP for SDNet" Xilinx, 2017
- [24] Jose Rolim, "Parallel and Distributed Processing: 11th IPPS/SPDP'99 Workshops Held in Conjunction with the 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing," San Juan, Puerto Rico, USA, April 12-16, 1999
- [25] P4. P4 to NetFPGA. <https://p4.org/p4/p4-netfpga-a-low-cost-solution-for-testing-p4-programs-in-hardware.html>
- [26] Han Wang, Robert Soulé, Huynh Tu Dang, Ki Suh Lee, Vishal Shrivastav, Nate Foster, Hakim Weatherspoon, "P4FPGA: A Rapid Prototyping Framework for P4," The Symposium on SDN Research, 2017.
- [27] Christoforos Kachris and Dimitrios Soudris, "A survey on recon gurable accelerators for cloud computing," In Proceedings of the International Conference on Field Programmable Logic and Applications (FPL). IEEE, 2016.
- [28] A. Beldachi, T. Diallo, K. Rajkumar, E. Hugues Salas, R. Wang, A. Tzanakaki, R. Nejabati, D. Simeonidou, "A Novel Programmable Disaggregated Edge Node Supporting Heterogeneous 5G Access Technologies", in Proc. ECOC, Dublin, Ireland, 2019.
- [29] B. Chatras, U. S. Tsang Kwong and N. Bihannic, "NFV enabling network slicing for 5G," *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, Paris, 2017, pp. 219-225. doi: 10.1109/ICIN.2017.7899415